Naval Research Laboratory

Washington, DC 20375-5320

Fuzzy Logic Resource Management and Coevolutionary Game-based Optimization

Jᴀᴍᴇs F. Sᴍɪᴛʜ III
Rᴏʙᴇʀᴛ D. Rʜʏɴᴇ II

Surface EW Systems Branch
Tactical Electronic Warfare Division

September 28, 2001

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
September 28, 2001	Formal	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Fuzzy Logic Resource Management and Coevolutionary Game-based Optimization	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
James F. Smith III and Robert D. Rhyne II	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Naval Research Laboratory Tactical Electronic Warfare Division Washington, DC 20375-5320	NRL/FR/5741--01-10001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR / MONITOR'S ACRONYM(S)
Naval Research Laboratory 4555 Overlook Ave., SW Washington, DC 20375-5320	11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A fuzzy logic expert system has been developed that automatically allocates electronic attack (EA) resources in real-time. This expertise-based resource manager is made up of four trees: the isolated platform tree, the multiplatform tree, the fuzzy parameter selection tree, and the fuzzy strategy tree. The initial version of the algorithm was optimized using a genetic algorithm using fitness functions constructed based on expertise. A new approach is being explored that involves embedding the resource manager in an electronic game environment. The game allows a human expert to play against the resource manager in a simulated battlespace with each of the defending platforms being exclusively directed by the fuzzy resource manager and the attacking platforms being controlled by the human expert or operating autonomously under their own logic. This approach automates the knowledge discovery problem. The theory of coevolutionary optimization is introduced, reoptimization criteria and stopping criteria are discussed, an algorithm for automatically constructing coevolutionary fitness functions is introduced, and examples are provided to show the effectiveness of coevolutionary optimization. A measure of effectiveness (MOE) for validation is discussed. Finally, the effectiveness of the resource manager and the optimization procedures is shown through a demanding example.

15. SUBJECT TERMS

Fuzzy logic, resource management, genetic algorithms, expert systems, distributed AI algorithms, knowledge discovery

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON James F. Smith III
a. REPORT	b. ABSTRACT	c. THIS PAGE	UL	27	19b. TELEPHONE NUMBER *(include area code)* 202-767-5358
Unclassified	Unclassified	Unclassified			

CONTENTS

FUZZY LOGIC RESOURCE MANAGEMENT AND COEVOLUTIONARY GAME-BASED OPTIMIZATION

1. INTRODUCTION

Modern naval battle forces generally include many different platforms, e.g., ships, planes, and helicopters. Each platform has its own sensors, e.g., radar, electronic support measures (ESM), and communications. The sharing of information measured by local sensors via communication links across the battlegroup should allow for optimal or near optimal decisions. The survival of the battlegroup or members of the group depends on the automatic real-time allocation of various resources.

A fuzzy logic algorithm has been developed that automatically allocates electronic attack (EA) resources in real-time. In this report, electronic attack refers to the active use of electronic techniques to neutralize enemy equipment such as radar [1]. The particular approach to fuzzy logic that is used is the fuzzy decision tree, a generalization of the standard artificial intelligence technique of decision trees [2].

The controller must be able to make decisions based on rules provided by experts. The fuzzy logic approach allows the direct codification of expertise forming a fuzzy linguistic description [3], i.e., a formal representation of the system in terms of fuzzy if-then rules. This will prove to be a flexible structure that can be extended or otherwise altered as doctrine sets, i.e., the expert rule sets change.

The fuzzy linguistic description will build composite concepts from simple logical building blocks known as root concepts through various logical connectives: "or," "and," etc. Optimization has been conducted to determine the form of the membership functions for the fuzzy root concepts.

The algorithm is designed such that when the scenario databases change as a function of time, the algorithm can automatically reoptimize, allowing it to discover new relationships in the data. Alternatively, the resource manager (RM) can be embedded in a computer game that EA experts can play. The software records the result of the RM and expert's interaction, automatically assembling a database of scenarios. After the end of the game, the software makes a determination of whether or not to reoptimize the RM using the newly extended database.

To be consistent with terminology used in artificial intelligence and complexity theory [4], the term "agent" is sometimes used herein to mean platform; also, a group of allied platforms is referred to as a "meta-agent." Finally, the terms "blue" and "red" refer to "agents" or "meta-agents" on opposite sides of a conflict, i.e., the blue side and the red side.

Section 2 briefly introduces the ideas of fuzzy set theory, fuzzy logic, and fuzzy decision trees. Section 3 discusses optimization with a focus on genetic algorithms. Section 4 discusses five major components of the RM. Section 5 examines coevolutionary theory, results, and software tools. Section 6 provides an example of the RM's response for a multiplatform scenario. Section 7 discusses a method of validating the resource manager and an algorithm that automatically invents new multiplatform EA techniques, rules, and strategies. Finally, Section 8 provides a summary.

2. A BRIEF INTRODUCTION TO FUZZY SETS, LOGIC, AND DECISION TREES

The RM must be able to deal with linguistically imprecise information provided by an expert. Also, the RM must control a number of assets and be flexible enough to rapidly adapt to change. The above requirements suggest an approach based on fuzzy logic. Fuzzy logic is a mathematical formalism that attempts to imitate the way humans make decisions. Through the concept of the grade of membership, fuzzy set theory and fuzzy logic allow a simple mathematical expression of uncertainty. The RM will require a mathematical representation of domain expertise. The decision tree of classical artificial intelligence provides a graphical representation of expertise that is easily adapted by adding or pruning limbs. Finally, the fuzzy decision tree, a fuzzy logic extension of this concept, allows easy incorporation of uncertainty as well as a graphical codification of expertise.

This section develops the basic concepts of fuzzy sets, fuzzy logic, and fuzzy decision trees. The parameterization of root and composite concepts is discussed.

2.1 Fuzzy Set Theory

This subsection provides a basic introduction to the ideas of fuzzy set theory. Fuzzy set theory allows an object to have partial membership in more than one set. It does this through the introduction of a function known as the membership function, which maps from the complete set of objects X into a set known as membership space. More formally, the definition of a fuzzy set [5] is as follows.

If X is a collection of objects denoted generically by x, then a fuzzy set A in X is a set of ordered pairs:

$$A = \{(x, m_A(x)) \mid x \in X\}.$$

$m_A(x)$ is called the membership function or grade of membership (also degree of compatibility or degree of truth) of x in A which maps X to the membership space M.

The logical connectives "and," "or," and the modifier "not" are defined as

$$or : A \cup B \rightarrow m_{A \cup B}(x) = \max[m_A(x), m_B(x)]$$
$$and : A \cap B \rightarrow m_{A \cap B}(x) = \min[m_A(x), m_B(x)]$$
$$not \quad B : \overline{B} \rightarrow m_{\overline{B}}(x) = 1 - m_B(x).$$

2.2 Fuzzy Decision Trees

The particular approach to fuzzy logic used here is the fuzzy decision tree. The fuzzy decision tree is an extension of the classical artificial intelligence concept of decision trees. The nodes of the tree of degree one, the leaf nodes, are labeled with what are referred to as root concepts. Nodes of degree greater than unity are labeled with composite concepts, i.e., concepts constructed from the root concepts [6] using "and," "or," and "not." Each root concept has a fuzzy membership function assigned to it. The membership functions for composite concepts are constructed from those assigned to the root concepts using fuzzy logic connectives and suitable modifiers. Each root concept membership function has parameters that are determined by optimization as described below.

Figure 1 offers an example of a decision tree. The logical connective "and" is denoted on the tree as a vertex with a line, the logical connective, "or" by a vertex without a line, and the logical modifier "not" as an edge with a circle through it.
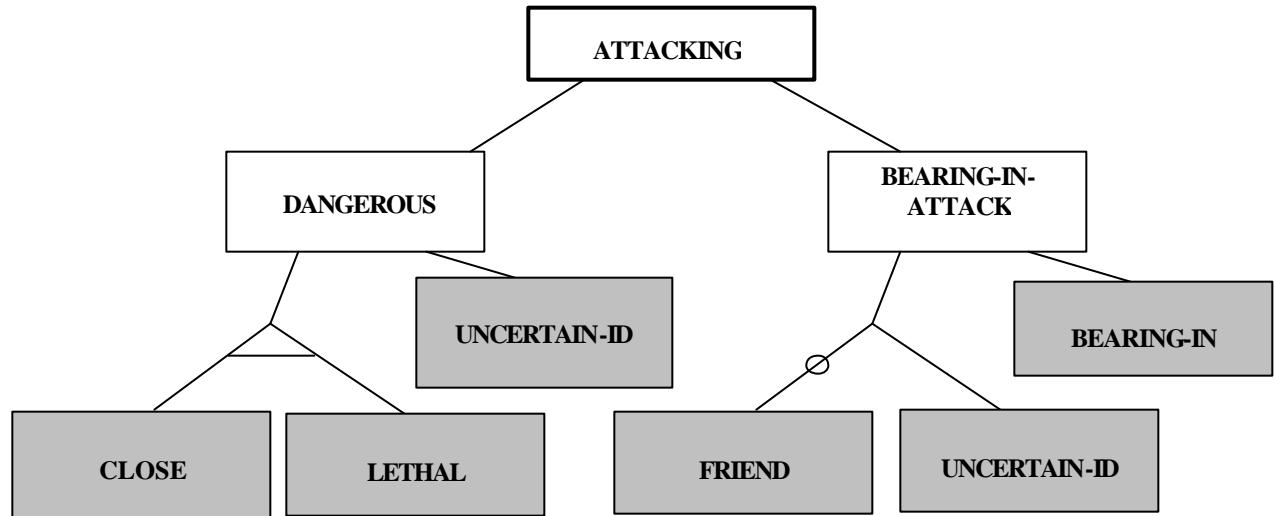
Fig. 1— A significant subtree of the isolated platform tree

3. OPTIMIZATION OF THE ROOT CONCEPT'S PARAMETERS USING A GENETIC ALGORITHM

The parameters of the root concept membership function are obtained by optimizing the RM over a database of scenarios using a genetic algorithm (GA). A genetic algorithm [7] can be described as follows. A genetic algorithm is an optimization method that manipulates a string of numbers in a manner similar to how chromosomes are changed in biological evolution. An initial population made up of strings of numbers is chosen at random or is specified by the user. Each string of numbers is called a "chromosome" and each numbered slot is called a "gene." A set of chromosomes forms a population. Each chromosome represents a given number of traits that are the actual parameters that are being varied to optimize the "fitness function." The fitness function is a performance index that is to be maximized.

The operation of the genetic algorithm proceeds in steps. Beginning with the initial population, "selection" is used to choose which chromosomes should survive to form a "mating pool." Chromosomes are chosen based on how fit they are (as computed by the fitness function) relative to the other members of the population. More fit individuals end up with more copies of themselves in the mating pool so that they will more significantly effect the formation of the next generation. Next, two operations are taken on the mating pool. First, "crossover" (which represents mating, the exchange of genetic material) occurs between parents. The final operation is mutation, which is the random change of a gene in a chromosome. After crossover and mutation, the next generation is formed, and the process is repeated until a stopping criterion is met [7].

The optimization procedures used here are a component of a knowledge discovery operation. Knowledge discovery or data mining is defined as the efficient discovery of valuable, non-obvious information embedded in a large collection of data [8]. The genetic optimization techniques used here are efficient, the relationship between parameters extracted and the fuzzy rules are certainly not a priori obvious, and the information obtained is valuable for decision-theoretic processes. Also, the RM is designed such that when the scenario databases change as a function of time, the algorithm can automatically reoptimize, allowing it to discover new relationships in the data.

The application of the genetic algorithm is actually part of the second step in a three-step data mining process. The first step is the collection of data and subsequent filtering by a domain expert to produce a scenario database of good quality. The second step involves the use of various data mining functions such as clustering and association. During this step, the genetic algorithm-based optimization is used to mine parameters from the database. These parameters allow the fuzzy decision tree to form optimal conclusions about resource allocation. In the third and final step of the data mining operation, a domain expert analyzes the RM's decisions to determine their validity.

Data mining to reoptimize the RM and real-time operation of the RM may occur simultaneously on different computers. Since the RM is designed to operate on a collection of platforms, even during very active use of the RM, some computer resources may be available for additional optimization and other data mining related activities. Thus, the multiplatform scheme allows frequent reoptimization of the RM, while the previously optimized version of the RM continues to function in real-time.

Typically the database is constructed from data taken from sensors of different types. The data will be sparse, intermittent, and noisy. To assemble a representative database, the domain expert must eliminate unacceptable data followed by the use of various data mining functions such as clustering [9-12] and association [13-20]. Clustering can be used for such tasks as organizing the data and suppressing outliers. Association determines when data measured on different sensors correspond to the same observable.

An alternate approach to constructing a database for reoptimization involves embedding the RM in a computer game. The game is designed so human EA experts can play it in real-time against the RM. The game software records the expert's selections. This record contributes to a database for reoptimization. Such a database is purer than one born of sensor data since such factors as environmental noise and sensor defects are not contaminating the data. This offers the advantage that the filtering stage of the data mining operation is simplified. The obvious disadvantage is that the database will be less representative of events in the real world than one born of real sensor data taken during battle.

4. THE SUBTREES OF THE RM

The resource manager is made up of four decision trees: the isolated platform decision tree (IPDT), the multiplatform decision tree (MPDT), the fuzzy parameter selection tree, and the fuzzy strategy tree. The EA decision algorithm, which can be called by the IPDT or the MPDT, is an expert system for assigning electronic attack techniques. The IPDT provides a fuzzy decision tree that allows an individual platform to respond to a threat [6]. The MPDT allows a group of platforms connected by communication links to respond to a threat in a collaborative fashion [6]. The communications model used for simulation purposes is described elsewhere [6]. The fuzzy parameter selection tree is designed to make optimal or near optimal selections of root concept parameters from the parameter database assembled during previous optimization with the genetic algorithm. Finally, the strategy tree is a fuzzy tree that an agent uses to try to predict the behavior of an enemy.

This section discusses the four major decision trees that make up the RM, the fuzzy EA decision algorithm and how they make efficient use of the Network-Centric paradigm. The Network-Centric paradigm refers to strategies that make optimal use of multiple allied platforms linked by communication, multiple resources distributed over different platforms, and decentralized command.

4.1 The Isolated Platform Decision Tree

The IPDT allows a blue platform that is alone or isolated to determine the intent of a detected platform. It does this by processing data measured by the sensors, e.g., ESM, radar, and IFF. Even when an incoming platform's ID is very uncertain, the IPDT can still establish intent based on kinematics.

When faced with multiple incoming platforms, the IPDT can establish a queue of which platforms to attack first.

Figure 1 (presented in Section 2) shows a significant subtree of the IPDT. The root concepts, those nodes of the tree of degree one, are represented as boxes with gray coloration. The other nodes or boxes are composite concepts.

The tree contains three classes of root concepts, all of which depend on measured information: those that make direct use of physics, those related to uncertainty in ID, and those related to information that is a function of ID and stored in databases. Some of the functions of ID considered are information-theoretic in origin. The root and composite concepts of the subtree of the IPDT depicted in Fig. 1 are similar to those found in the sensor management literature [2]. They differ in fuzzy membership functions, interpretation, and application. Additional concepts on the IPDT not found in the literature will be the subject of a future publication.

4.1.1 Root Concepts Related to Physics

The root concepts "close" and "bearing-in" belong to the first class, i.e., they are directly related to physics. The root concept "close" has been described in detail elsewhere [6, 21-22]. Bearing-in uses the same membership function as "close" with range replaced by bearing.

Each root concept fuzzy membership function is dependent on a physical observable O and frequently, its first derivative in time, dO/dt. If the root concept is only dependent on O, then the membership function together with a threshold defines an interval, such that if a measured value of O falls outside this interval, the membership function takes on a certain value that may trigger an "action" by the RM. The two-dimensional space resulting from plotting dO/dt vs O is a phase space. The inequality between the root concept membership function and its threshold upon inversion will give inequalities in O and dO/dt, typically. The resulting system of inequalities defines a region of phase space referred to as the admissible region where red can engage in activities without signaling its intent to blue. The membership function parameters that are found through data mining determine the boundaries of the admissible region of phase space. The admissible region cannot in general be brought to zero area; otherwise, blue will carry out an action against everything it detects, resulting in fratricide and wasting valuable resources essential to its survival.

4.1.2 Root Concepts Related to ID

Just as quantities related to geometry and kinematics such as range, bearing, and elevation are all inputs to the IPDT, it is assumed that an ID classification is also an input. The ID is represented as a classification vector. The three ID subclasses making up the ID classification vector are friend_type, neutral_type, and foe_type. Ideally, these would be non-fuzzy or crisp concepts, i.e, the ID of an incoming platform would be certain as to if it is a friend, neutral or foe. The RM has the ability to deal with uncertain ID, so each of the ID subclasses corresponds to a fuzzy set of the same name and each incoming platform has a fuzzy degree of membership in each of these fuzzy sets. This generally proves to be a very valuable approach and even in the case of very good ID information this formalism still is very effective since relevent grades of membership can be assigned values of unity or zero.

Each time new input data are provided to the RM by sensors, the ID information is provided in the following format. The RM at input update time t is provided with the ID uncertainty vector $\vec{U}(i,t)$, whose elements are the grade of membership of the ith emitter in the fuzzy subsets for friend_type, neutral_type, and foe_type, i.e.,

$$\vec{U}(i,t) = (m_{friend}(i,t), m_{neutral}(i,t), m_{foe-type(1)}(i,t), m_{foe-type(2)}(i,t),..., m_{foe-type(n)}(i,t)),$$

where the subscripts *foe-type(1), foe-type(2), …, foe-type(n)*, indicate that there can be *n* foe types that can threaten a blue platform. The elements of the ID uncertainty vector are defined such that their sum is less than or equal to unity. The RM can also deal with input relating to more than one friend-type and neutral-type, but that is beyond the scope of this discussion.

4.1.3 Root Concepts that are a Function of ID

It has also proven valuable to define additional root concepts that are a function of the fuzzy grades of membership of the elements of the ID classification vector. A useful concept that is a function of ID is "lethal." The membership function for the concept lethal is defined in terms of the sum given below:

$$m_{lethal}(i,t) = \sum_{j=1}^{n} m_{foe-type(j)}(i,t).$$

Given the grades of membership for "lethal," "friend," and "neutral," the fuzzy membership function for the concept "unknown" is defined as follows:

$$m_{unknown}(i,t) = 1 - [m_{lethal}(i,t) + m_{friend}(i,t) + m_{neutral}(i,t)].$$

This is the fuzzy complement of the sum of the elements of the ID uncertainty vector. The concept "unknown" quantifies the the degree to which the ID of the incoming emitter is unknown.

One frequently used measure of global uncertainty is the fuzzy entropy [5], which is defined as

$$S = -\sum_{k} m_k(i,t) \ln m_k(i,t).$$

To apply the fuzzy entropy as a measure of global uncertainty for ID, it is useful to make the following definition,

$$m_k(i,t) = \begin{cases} m_{friend}(i,t) & k=1 \\ m_{lethal}(i,t) & k=2 \\ m_{neutral}(i,t) & k=3 \\ m_{unkown}(i,t) & k=4. \end{cases}$$

The root concept "uncertain-ID," which provides a measure of global uncertainty in ID, is defined as

$$m_{uncertain-ID}(i,t) = -K \sum_{k=1}^{4} m_k(i,t) \ln m_k(i,t),$$

with *K* a constant defined so that the fuzzy membership function does not exceed unity.

4.2 The Multiplatform Decision Tree

The IPDT made limited use of the Network-Centric paradigm, using the other networked platforms for surveillance and electronic intelligence. However, the purpose of the Network-Centric paradigm is to use multiple platforms to gain geometric, physical, and tactical advantage by using multiplatform techniques that are more effective than standard techniques. Such techniques require coordination and communication from platform to platform, as well as some command and control structure.

4.2.1 Platform-to-Platform Interactions

The IPDT allowed an isolated platform to respond to an incoming emitter. The RM running on the isolated platform based its decisions and, hence, response on standard sensor output, e.g., range, range-rate, and bearing. The isolated platform's response can range from simply continuing to monitor the environment to deciding to engage in EA. If a decision to engage in EA is made by the RM, a call is made to the fuzzy EA decision algorithm, which is discussed in Section 4.3.

As it stands, the IPDT cannot take full advantage of the Network-Centric paradigm. To do this, another decision tree, the MPDT, is required. Using sensor output, the MPDT allows a group of platforms, connected by a communications network to work together in an optimal fashion, to take advantage of the full potential of the Network-Centric paradigm.

Figure 2 depicts a significant subtree of the MPDT. The MPDT required many new rules, some analogous to rules found on the IPDT, but most quite distinct. The following will examine at a coarse level some of these rules and their related fuzzy concepts.
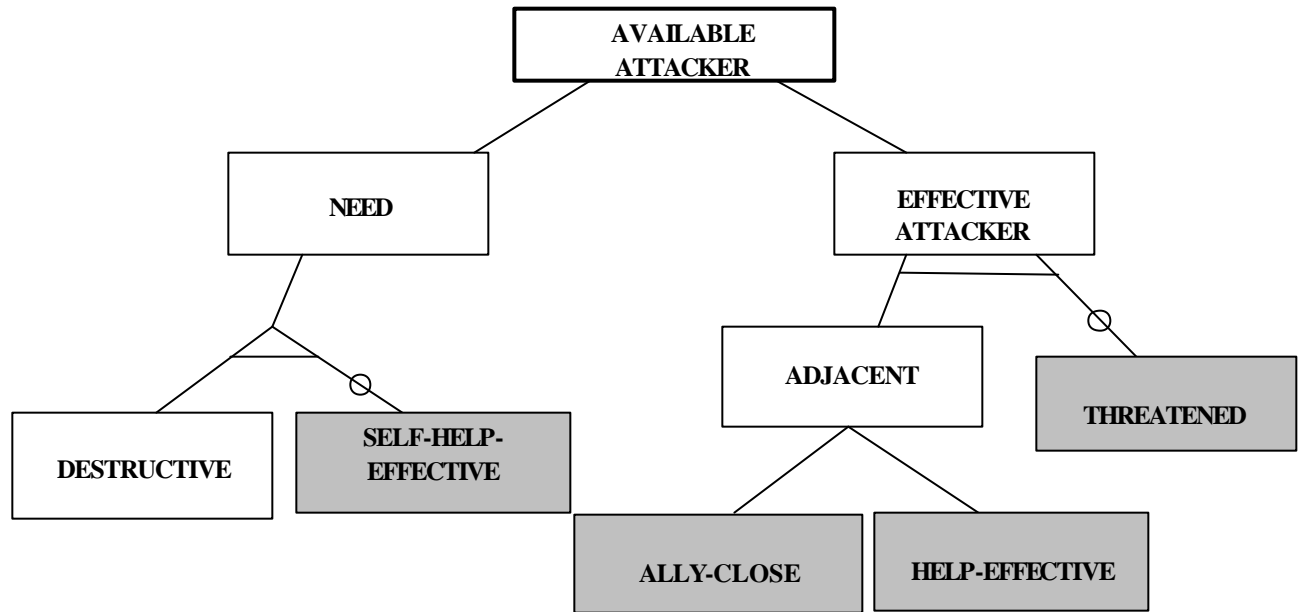
Fig. 2 — A significant subtree of the multiplatform decision tree

4.2.2 Some Root and Composite Concepts on the MPDT

The first rule to be defined is the fuzzy concept of a platform's need. If the RM aboard a blue platform determines a threat is "attacking" by using the IPDT, then the detector should alert other platforms to its "need" for assistance.

A platform's "need" is a function of its ability to respond to a threat and how destructive the threat is perceived to be. The composite concept "need" is constructed using the membership functions for the root concepts "self-help-effective" and "destructive," as shown in Fig. 2. The membership function for "self-help-effective" is a function of the EA resources aboard the ith platform, where "need" is being determined.

The composite concept "destructive" is constructed from the root concepts "potentially destructive" and "kinetic-energy destructive" (not pictured). The fuzzy membership function for "potentially destructive" is actually an index between zero and one, assigned by experts detailing how threatening the emitter is perceived to be in terms of its onboard hardware. The fuzzy membership function for "kinetic-energy-destructive" is a function of the emitter's estimated translational and rotational kinetic energy. In actual application there are other root concepts contributing to "destructive," the concept has been simplified here due to space limitations.

The composite concept of "need" reduces the amount of data that has to be sent over the network. It does this by sending processed information, as opposed to raw data, over the network.

The composite concept "adjacent" checks platform/threat disposition, along with resources onboard the potential "helper" platform. A helper platform is one that is not threatened but has received a communication message that another platform is threatened, i.e., the threatened platform is communicating to the helper that it has "need." The fuzzy root concept "ally-close" relates to how close the threatened ally is to the platform that is evaluating its ability to help in terms of the concept "adjacent." The root concept "help-effective" relates to how effective the helping platform might be if it should come to the assistance of the threatened platform that has "need."

The composite concepts "effective attacker" and "need" are combined through an "and" connective to construct the composite concept "available attacker." If the membership function for "available attacker" exceeds a certain threshold, the helping platform comes to the assistance of the platform with need. Note that the parts of the tree leading up to "need" are calculated on the threatened platform; the subtree for "effective attacker" and the final "and" operation between "need" and "effective attacker" are calculated on the helping platform. This allows the RM to take advantage of multiple computers within the blue platform group.

4.3 The Fuzzy EA Decision Algorithm

Once the IPDT or the MPDT determines an action is required, the fuzzy EA decision algorithm becomes active. This fuzzy algorithm allows the RM to pick the best EA technique(s) to use against the incoming emitters. The RM's decision is based on the emitters' ID, uncertainty in ID, available assets within the blue group, blue asset reliability, logistics for resupplying blue assets, battlespace geometry, intelligence reports related to red assets, red asset reliability, logistics for resupplying the red forces, weather and atmospheric conditions, etc.

The fuzzy EA decision algorithm is an expert system based partially on military doctrine obtained by interviewing experts, preferred techniques found in the literature, and entirely new classes of techniques invented to exploit the Network-Centric paradigm.

A new class of algorithms has been developed specifically for the purpose of inventing optimal EA techniques for use under the Network-Centric paradigm. These are referred to as combinatorial EA algorithms. The name combinatorial EA derives from the approach. For a given military scenario, the various blue and red platforms and their assets are simulated. Each platform in the blue platform group can use certain EA techniques. The different combinations of techniques over the blue platform group are enumerated and stored as a vector; each element of the vector corresponds to a technique a blue platform in the group can use. The value of using such a technique combination is determined using a measure of effectiveness (MOE). The number of possible combinations of techniques for multiplatform engagements is typically large, so an efficient search algorithm is required. The current approach to search and evaluation uses a genetic algorithm with a modified version of the MOE acting as a fitness function.

When blue selects an EA technique or a combination of EA techniques, there is always the danger that red will make use of a countermeasure designed to circumvent blue's deception. This probability of

red's success against blue can increase if blue uses the same EA techniques each time similar red strategies are encountered.

Two algorithms that automatically invent new multiplatform EA techniques, rules, and strategies are under development. They are discussed in greater detail in Section 7.

4.4 The Fuzzy Parameter Selection Tree

The fuzzy parameter selection tree can be called by the IPDT, MPDT, the fuzzy strategy tree, and the fuzzy EA algorithm. For each tree, it allows the selection of the best parameters determined offline using genetic optimization. The selections are a function of such data as emitter ID, uncertainty in ID, intelligence reports, battlespace geometry, geography, and weather. These parameters can include probabilities for the best strategy calculated using game theory.

By selecting specialized parameter sets for different situations, the RM can use the same decision trees and functional forms for the fuzzy membership functions. This also allows the RM to be used on many different types of blue platforms and deal with very general red threats.

4.5 The Fuzzy Strategy Tree

A strategy tree is an agent's concept of an opposing agent's decision tree. If an agent has sufficient knowledge of the enemy's past behavior, the strategy tree can be very useful for predicting future behavior. To make this idea more concrete, consider the root concept "close" on the blue decision tree. This root concept deals with the ideas of how near the red platform is to the blue platform and how fast the red platform is approaching the blue platform. If red is near or approaching fast, the close membership function will assume a value near one. If the membership function exceeds a certain threshold, even though red's ID information has a high degree of uncertainty, the RM will engage in an action, i.e., it will execute EA against the incoming red platform. Red desires to get sufficiently near blue so that a red action can occur. If red knows the mathematical form of the membership function for "close" exactly and has approximate knowledge of the related parameters, then red can get very near to blue without triggering an action. Red's version of "close" on his strategy tree, together with the associated threshold and the grade of membership that it must not exceed to avoid an action by blue, determines inequalities. These inequalities determine a region of the range-rate vs range-phase space, referred to as the admissible region of phase space as defined in Section 4.1.1. If red remains in that region and red's parameters for "close" depart little from those of blue, then red will not alert blue as to his intention. Thus red can use a strategy tree to get very near blue before executing an action.

More generally, as described in Section 4.1.1, each root concept gives rise to an admissible region of phase space. If red's ID is highly uncertain, and through his strategy tree red has a good notion of the geometry of each admissible region of each phase space associated with each of blue's root concepts, then red can exploit this knowledge to greatly increase his likelihood of beating blue. Just as with red, blue typically has a strategy tree and uses it to exploit knowledge of red's past behavior.

5. COEVOLUTION AND SOFTWARE TOOLS

This section discusses coevolutionary data mining and its relation to strategy tree theory, the stopping criteria for coevolutionary data mining, the reoptimization criteria, and an algorithm for automatically constructing the coevolutionary fitness function. Software tools are discussed that allow the full solution of the data mining problem using the scenario generator. Finally, a detailed example for the root concept "close" is given.

5.1 Coevolution

In nature, a system never evolves separately from the environment that contains it. Both biological system and environment evolve simultaneously. This is referred to as coevolution [23-27]. Similarly, the fuzzy resource manager should not evolve separately from its environment, i.e., enemy tactics should be allowed to evolve simultaneously. Certainly, in real-world situations, if the enemy sees the resource manager use a certain range of techniques, the enemy will evolve a collection of counter techniques to compete more effectively with the resource manager.

A previous report [6] explored an approach to coevolution involving averaging over a database of military scenarios. The current approach involves both blue and red meta-agents, each having fuzzy decision trees and strategy trees. Both types of tree will adapt during optimization. A strategy tree differs from a decision tree in that it is one meta-agent's model of another meta-agent's decision tree. During coevolution, as a meta-agent "learns" the behavior of its enemy, the parameters in its strategy tree will be adjusted, finally duplicating those in the enemy meta-agent's decision tree.

5.2 The Strategy Tree Approach to Coevolutionary Data Mining

The approach to coevolution is as follows. A threshold is defined for each root concept membership function on the red strategy tree, such that if the membership function exceeds this threshold and if red's strategy tree is a good representation of blue's decision tree, then red's intention is signaled to blue, resulting in an action by blue. The membership function is typically a function of some physically measurable quantity O and its first derivative in time, dO/dt. The two-dimensional space resulting from plotting dO/dt vs O is a *phase space*. The inequality between the root concept membership function and its threshold, upon inversion, will give inequalities linear in O and dO/dt, typically. The resulting system of inequalities defines a region of phase space referred to as the *admissible region* where red can engage in activities without signaling its intent to blue. The membership function parameters that are found through data mining determine the boundaries of the admissible region of phase space. The admissible region cannot in general be brought to zero area; otherwise, blue will carry out an action against everything it detects, resulting in fratricide and wasting valuable resources essential to its survival.

The concept "close" refers to how near the target/emitter on track i is to the ship, or more generally the platform of interest [22]. The universe of discourse will be the set of all possible tracks. Each track i has membership in the fuzzy set "close" based on its range R (nmi) and range rate dR/dt (ft/s). The fuzzy membership function for "close" takes the form

$$m_{close}(i) = \frac{1}{1 - a \mid R_i - R_{min} \mid / \max(-\dot{R}_i, \dot{R}_{min})} .$$

The parameters to be determined by optimization are

$$a, \quad R_{min}, \quad \text{and} \quad \dot{R}_{min} .$$

The parameters for "close" were initially determined using a genetic algorithm [21, 22]. The fitness function used for initial optimization (i.e., before the beginning of the coevolutionary process) is described in Refs. 21 and 22. This fitness function is the zeroth order fitness function for coevolution.

The gray region of Fig. 3 is the admissible region of the range-rate vs range-phase space determined using the above procedure for the root concept "close" on red's strategy tree. It is assumed red knows the exact mathematical form of blue's fuzzy membership function for "close," but red only knows the parameters for "close" approximately. Quantities with an "*r*" superscript are those assumed by red, "*i*" subscripts refer to the ith track, the k subscripts indicate values at the kth time step, and "*0*" subscripts are

used on red's initial values of range and range-rate. The symbol τ^r is the threshold that red's grade of membership in "close" should not exceed so as to not signal red's intent to blue. The quantity d_H is the desired distance red would like to be from blue before executing an action.

Once the admissible region of phase space is determined it is straightforward to find trajectories for red that are optimal according to some criterion and at the same time allow red to approach blue without indicating its intent. For example, the trajectory that allows red to spend the minimum amount of time in the admissible region of phase space is collinear with the line segment determined by points A and B in Fig. 3. The associated acceleration is

$$\ddot{R}^r_{i,k} = -(R^r_{min} - R^r_{i,k}) \cdot \left(\frac{a^r t^r}{1 - t^r}\right)^2$$

This trajectory allows red to travel at the maximum absolute range-rate at each time step. It is also a high risk trajectory for red, since if red has been even slightly overly optimistic about blue's parameters, red will depart from the admissible region and signal its intent to blue.
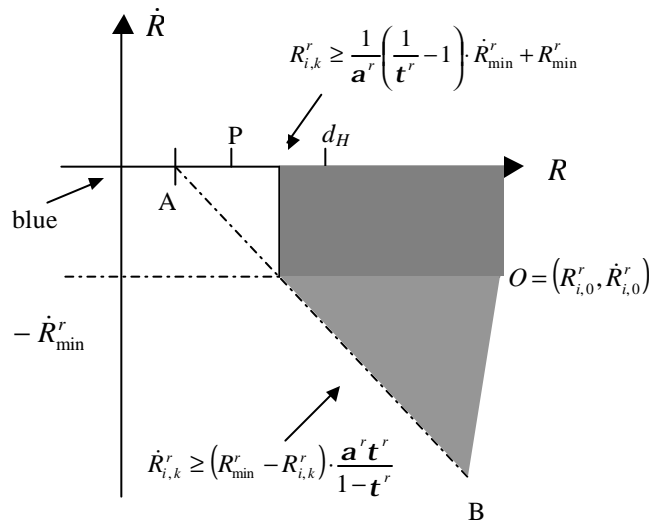

Fig. 3 — The gray shaded area is the admissible region of phase space that red seeks to occupy

It is important to realize that the fitness functions used for initial optimization, i.e., prior to the beginning of coevolution, are born of expertise. The fitness functions are highly nonlinear and lack derivatives at many points. The resulting optimization problem is a natural candidate for application of a genetic algorithm. When it can be established that only small changes in parameters are required, then faster approaches to reoptimization can be used.

5.3 Tools for Visualization of Data-Mined Information

To facilitate data mining, coevolution and validation of the RM, a software tool known as the scenario generator (SG) has been created. It automatically creates simulated blue and red platforms with user-defined assets. It also creates a map or battlespace and automatically places the red and blue platforms in this space where they can interact. Each blue platform is controlled by its own copy of the fuzzy RM.

The SG has two modes of operation. In the computer vs computer (CVC) mode, each red platform is controlled by its own controller distinct from the fuzzy RM used by the blue platforms. In the second mode, the human vs computer (HVC) mode, a human player controls a red platform through an interactive graphical user interface (GUI). There can be multiple red platforms. At each time step, the human player can control any of the red platforms, but only one of them per time step. Those red platforms not under human control run under their own logic as in the CVC mode.

Three different GUIs can be easily accessed from the SG software. These GUIs are the "scenario builder," the "map builder," and the "human control player interface" (HCPI).

Figure 4 displays the scenario builder GUI that allows the construction of blue and red agents with general characteristics. Through this GUI both blue and red agents can be given various assets such as different types of radars, ESM, EA systems, etc. This GUI allows the creation of a terrain map that is discussed below. The scenario created can be placed in a database for further data mining and coevolutionary analysis. Effects due to weather, system losses, atmospheric attenuation, multipath, clutter, etc., can be included in the calculation, although they can not be currently called from the GUI.
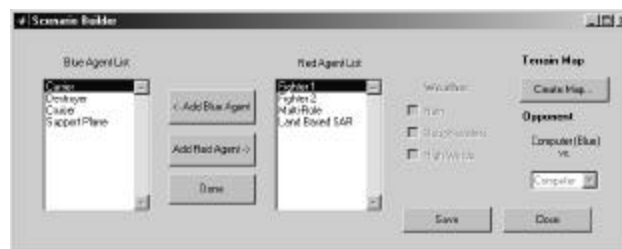

Fig. 4 — "Scenario Builder" tool

Figure 5 displays the map builder GUI that can be called from the scenario generator GUI. The map builder allows the construction of various maps on which the red and blue agents can interact as the scenarios are played out. The map defines a battlespace that can include various environments such as oceans, forest, deserts, cities, and jungles. Maps created by the map builder can be saved in a database for reuse.
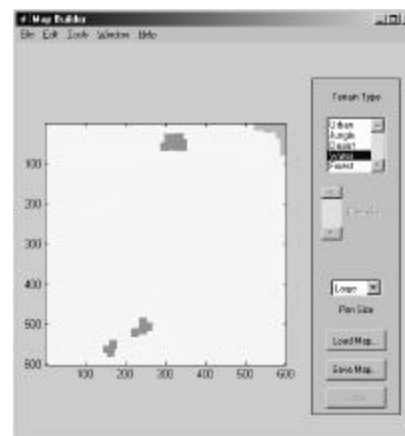

Fig. 5 — "Map Builder" tool

Once the above tools have been used to create blue and red platforms, their assets, and the battlespace map, a battle can be initiated in either CVC or HVC modes. In both modes, the results are recorded in the database for later data mining to improve the RM's adaptive response. Section 6.2 discusses an example of output from the SG running in CVC mode.

Figure 6 displays the human control player interface that is used in the HVC mode. It includes a digital simulation of a radar's PPI display similar to those used in real radar systems. Target range and bearing can be determined from the PPI display or the digital readout at the lower right. When the blue target is in range, the human player can fire a missile. A probabilistic model determines the effectiveness of the missile. Finally, each red platform has a limited number of missiles so the player must be cautious in using them.
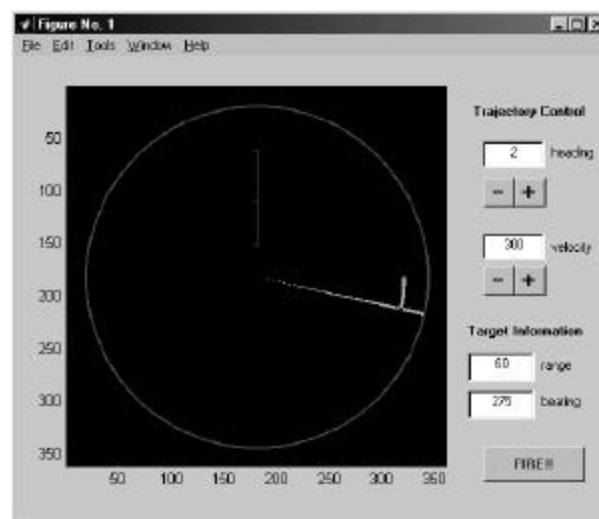

Fig. 6 — The human control interface that allows a human EA expert to control a red platform

The simulated PPI display is designed to imitate the properties of a real PPI display. In this way, decisions made by the EA expert playing the game that are subsequently automatically stored in the database for later data mining will reflect truth. The display can also show the effects of naturally occurring noise, radar system noise, multipath, clutter, jamming, and false targets, and can simulate the phosphor decay of a real radar PPI display.

5.4 Criterion for Reoptimization

The criterion for reoptimization is formulated based upon a determination that a particular parameter set has become ineffective. In the EA community, failure to delay, disrupt, or deny information can be the basis for labeling a parameter set ineffective. Outright loss of platforms is another simple measure of ineffectiveness that can be used. Kindred to how humans change and replace strategies, the RM uses the reoptimization criterion as a point of rebuilding and reconstructing strategies. When the reoptimization criterion is triggered, a GA is called to reoptimize the root concepts of the decision and strategy trees.

One simple approach to determining this criterion is to analyze root concept membership functions' values over time when blue loses to red. If it is determined that blue's failure related to certain root concept membership functions not triggering an action by the RM, these membership functions could be made more sensitive.

After many coevolutionary generations, it is possible that both the blue and red groups will have evolved to the point that they are very effective in dealing with each other but no longer effective in dealing with agents from past generations. For a real system running the RM, this could be a deadly defect, as it is not uncommon to encounter opposing systems manufactured at many different times.

5.5 Stopping Criterion for Coevolution

Just as with a genetic algorithm, in a coevolutionary game-based optimization, a stopping criterion must be defined. Upon completion of the game, several iterations of a particular scenario can be played. A criterion for reoptimization determines when the RM will reoptimize its parameters. This optimization is kindred to the scenario-based optimization discussed in Refs. 6 and 20, however the scenarios optimized over are recordings of the previous games since the last optimization. Since the reoptimization criterion determines how many scenarios the optimization is taking into account, this criterion is nontrivial.

5.6 Automatic Construction of a Fitness Function for Coevolution

When reoptimizing it is necessary to incorporate knowledge of an agent's history, specifically those events that led to reoptimization. A method of doing this is to construct fitness functions that contain a history of the agent and upon maximization result in agents that will not reproduce past mistakes. This subsection develops an algorithm for the automatic construction of such functions, referred to as symbolically recursive fitness functions.

This first step in producing a symbolically recursive fitness function involves multiplying the fitness function used in the previous coevolutionary generation for blue optimization by a product of Heaviside step functions for the current coevolutionary generation. The fitness function for the first coevolutionary generation is formed by multiplying the Heaviside step functions by the fitness function used during the initial genetic algorithm-based data mining process, referred to as the zeroth order fitness function.

The product of Heaviside step functions includes one Heaviside step function as a factor for each offending root concept evaluated at each time step since the last reoptimization. The argument of each Heaviside step function is the difference between the offending root concept membership function evaluated appropriately at each time step and a threshold. The resulting product fitness function is referred to as a symbolically recursive fitness function. The idea is that unless the GA optimization produces a root concept membership function which for this set of input data, exceeds the appropriate threshold, the symbolically recursive fitness function will return a value of zero. Using the symbolically recursive fitness function the GA can be used to ensure that the membership function value for a particular root concept will be above a certain threshold. This triggers an appropriate action by the RM the next time red exhibits the behavior that led to blue's loss and subsequent reoptimization. A similar procedure is used when reoptimizing red.

By evaluating the fitness over the current coevolutionary generations as well as previous coevolutionary generations, the resulting parameter sets will be effective for the current red strategy, as well as the previous red strategies. This allows the RM to adapt to current strategies without being vulnerable to previous strategies that could be used by older red agents.

5.7 A Simple Example of Coevolutionary Optimization Using the Fuzzy Concept "Close"

This subsection provides a simple example of coevolutionary data mining using the fuzzy root concept "close." CVC and HVC coevolution are considered as well as a comparison between the techniques.

For both HVC and CVC modes, a loss by blue results in immediate reoptimization of blue's parameter set. A loss by a computerized red agent results in reoptimization of the red agent's parameter set. The stopping criterion for reoptimization for both modes is a maximum number of coevolutionary generations. A coevolutionary generation refers to a single battle followed by reoptimization of red or blue.

A blue loss occurs if one of blue's agents is disabled due to the successful delivery of a red missile. A probabilistic model determines the effectiveness of the fired missile. A blue win occurs if the blue agent group is able to delay red a certain number of time steps t. Finally, a red loss occurs if blue wins.

In HVC mode, the human player acting as a red agent can locate blue agents using the PPI display described in Section 5.3. When a blue agent is located on the screen, the user clicks on the target region and presses the fire button located in the lower right-hand corner to launch a missile.

One simple class of experiments that has been conducted consists of one blue agent vs one red agent. It was typically found that all three parameters in blue's version of "close" showed little change for the last 33% of the coevolutionary generations. The human opponent operating the red agent tended to fixate on the same strategies. This suggested that in HVC optimization the human player quickly reached the limits of his or her expertise, resulting in the RM's parameters reaching a constant value. Thus the optimization of blue converged rapidly.

In CVC mode, there is no human player controlling a red agent. The red agents are controlled by their own logic that includes a strategy tree. Each blue agent is controlled by a copy of the RM, as in HVC mode. The blue agent's decision tree has the root concept "close" on it. The red agent has a strategy tree with his perception of "close." It is assumed that red has very good intelligence about blue, hence the mathematical form that red is using for "close" is the same as the one blue uses. Red is uncertain about the value of blue's parameters for "close" and, as such, how they slightly differ from those of blue.

Both blue and red can change during the coevolutionary process. Red's parameters for his version of "close" determine the admissible region of phase space that red attempts to occupy so as not to invoke an action by blue. Assuming a given initial position and velocity for red, these parameters in turn determine red's value of acceleration, hence red's trajectory.

In a simple experiment with one blue agent vs one red agent operating in CVC mode, convergence was not nearly as fast as in HVC mode. The computer-controlled red agent is typically capable of exhibiting many more strategies than the human-controlled red agent in HVC mode. Thus the coevolutionary process ends up exploring the combined red-blue parameter space longer, resulting in a greater likelihood of a global maximum being found for the fitness function, resulting in an RM that is more robust than in the HVC case.

The more robust RM obtained through use of the CVC optimization can be understood intuitively as follows. If red can exhibit more strategies by using CVC mode than in HVC mode, then the blue RM is forced to be more adaptive to compete.

There is a risk during coevolution that with both red and blue coevolving, they will become very specialized in dealing with each other. For example, without taking proper precautions, blue agents of the 1000th coevolutionary generation might be effective against red agents of that generation, but ineffective against agents of generations 100 through 999. Fortunately, the structure of the symbolically recursive fitness function prevents this, because its form retains the past history of the agents, forcing the blue agents of the 1000th generation to be effective against red agents of the preceding or current generation.

6. EXAMPLES OF MULTIPLATFORM RESPONSE

This section examines a specific example of the fuzzy RM's ability to optimally allocate electronic attack resources. Input requirements and output characteristics are considered and illustrated through the actual output of the current implementation of the RM. Many examples like those included in this section demonstrate that the information data-mined using the GA is extremely valuable.

6.1 Input Scenarios and Output of the Fuzzy RM

The fuzzy RM uses as input the position and number of ally platforms (ships, planes, etc.), as well as emitter range, bearing, elevation, and the emitter ID, with the associated uncertainty for the ID. The effect of the data is to stimulate the various fuzzy logic concepts, resulting in different "actions" by the algorithm. The emitter ID is used to determine the technique or techniques (for ID's with uncertainty) that the ally platform or platforms can execute against the emitter.

Figure 7 shows a battleforce of three ships and also an incoming aircraft with counter targeting (CTAR) radar. The type of the threat emitter is not well known, i.e., the aircraft's ID is not known with 100% certainty. The ship initially closest to the airplane, the carrier, is disabled and cannot participate in joint EA. With the threat's classification not being well known, and because a foe of some type is indicated, the RM directs the two ships with functioning EA systems to engage in joint EA against the incoming foe, subsequently defending the disabled ship.

Midway through the battle, a helicopter was detected by the ships' sensors. It was determined to be a foe with uncertain radar ID. The fuzzy RM determined joint EA was called for and directed the two ships with functioning EA systems to use two beams and simultaneously conduct joint EA against the incoming airplane and helicopter.

Fig. 7 — Input scenario for airplane threat with
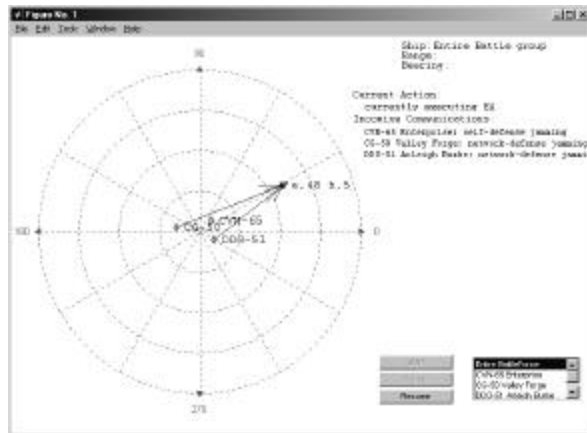uncertain radar ID, late arriving helicopter threat

Fig. 8 — RM output for the first part of the scenario where two platforms are attacking the threat with the third platform disabled
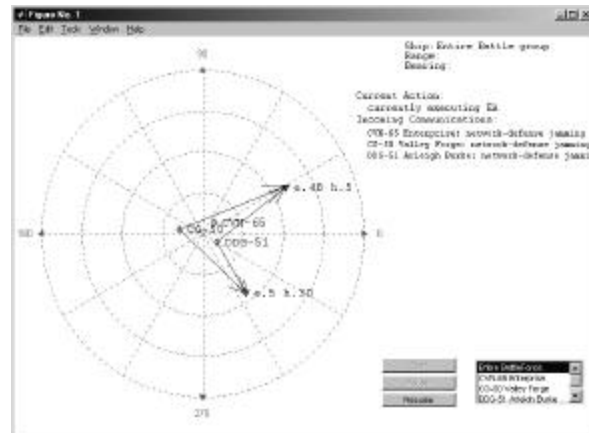
Fig. 9 — RM output for the second part of the scenario where both platforms split their beams to attack the late arriving enemy helicopter

Figure 8 displays the output of the algorithm during the first part of the scenario depicted in Fig. 7, when only the airplane has been detected. A polar plot with origin at the centroid of the battlegroup is used to display the positions of the three ships (diamonds), and the incoming emitter (triangle marked with designation "foe type"). Communications and electronic attack techniques used by each ship are listed to the side. The arrows running from the ships to the foe-type emitter indicate electronic attack. The arrows indicate that the RM has directed the two ships with functioning EA systems to attack the airplane since it has been determined to be a foe with uncertain radar ID. The carrier continues to monitor the battlespace.

Figure 9 displays the output of the RM during the final stage of the scenario during which the second foe, the helicopter, was detected by the sensors. Since the helicopter was a foe with uncertain radar ID, the fuzzy RM decided it should be attacked. The RM directed the two ships with functioning EA systems to use two beams and simultaneously attack both foes.

A copy of the RM runs on each of the blue platforms. A polar plot of the kind depicted in Figs. 8 and 9 is displayed each second of the RM's operation. The RM makes its decisions as a function of battlespace geometry, blue assets, and intelligence reports related to red assets. At any given time, there is no commanding platform. This is a valuable aspect since if a blue platform is lost through any mechanism, the group is not delayed as command is transferred to another platform.

The RM has been tested for many different military scenarios [6,21-22,24-27]. It has been determined to be very effective by comparing its decisions to the judgment of human experts.

6.2 A Battle Created Using the Scenario Generator

The software described in Section 5.3 is extremely useful for evaluating the RM and determining the value of information data-mined in the second data mining step. The natural output of the scenario generator is a computer-generated movie. This subsection includes frames from such a movie for the scenario described below. It illustrates the operation of the RM while the SG runs in the CVC mode. The scenario generator also creates a corresponding database reflecting the RM's decisions for later analysis and subsequent data mining to improve the RM's adaptive response.

The following are the events leading up to the fictitious battle. A blue plane is downed and a blue platform group is sent in to rescue it. The blue group consists of four ESM-/EA-equipped ships as follows: one carrier, one cruiser, and two destroyers. There is also a blue rescue helicopter and an ESM-/EA-equipped blue support plane. The blue group will encounter a threat mix consisting of three red fighter planes (each with multirole radars) and two red land-based search and acquisition radars (LSAR). Due to the geopolitical diversity of the region, IDs are only given with 50% certainty. The RM handles uncertainty in ID very well.

Figures 10 through 13 display the simulation created by the scenario builder and map builder for time steps 1, 3, 7, and 11. Darker regions indicate desert and lighter regions water. The particular time step that the picture corresponds to is given at the bottom in the left-hand corner. Each platform is indicated by an asterisk labeled with the platform's type. The platform's activity at each time step is displayed next to its type. A blue platform's jamming process is depicted as a line emerging from the blue platform and ending on the red platform. Figure 11 depicts the first instance of jamming.
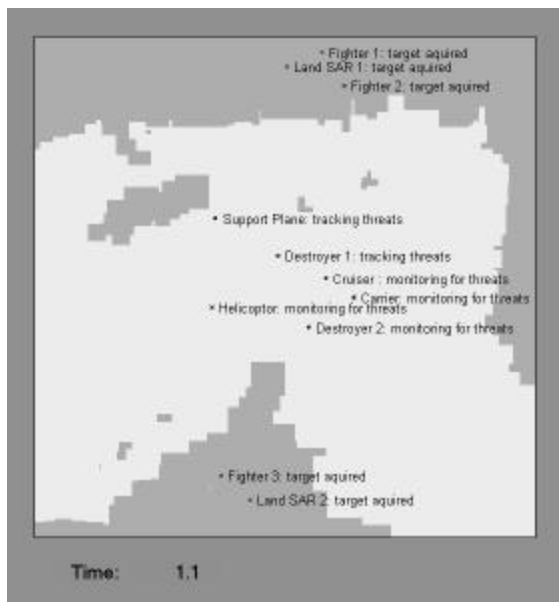

Fig. 10 — The first time step of the blue RM's operations; the six blue platforms monitor for red activity
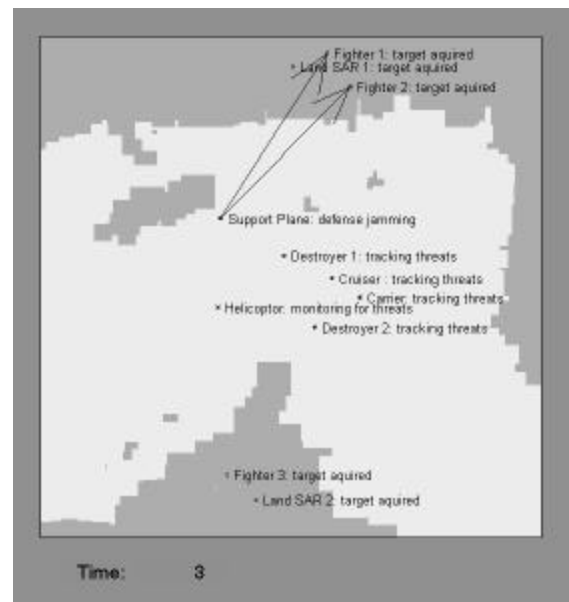

Fig. 11 — The third time step of the blue RM's operations, blue sensors have detected the presence of red radar activity. The RM has directed the blue support plane to engage in EA against two red platforms.

Figure 10 depicts the initial state of both the red and blue platform groups during the first time step of the RM's run. A copy of the RM running on each blue platform determines its behavior. An algorithm different from the blue RM controls the red platforms. The events of the battle are saved in a database for data mining operations and coevolutionary analysis.

During the first time step, the southern red land-based search and acquisition radar acquires the blue rescue helicopter. The southern LSAR communicates this information to the rest of the red group. The blue helicopter is subsequently acquired by the remaining red platforms.

Figure 11 depicts the events of time step 3. The northern LSAR has acquired prior to this time step the blue support plane and destroyer 1. The blue group determines, based on sensor data, that CTAR systems are active. The fuzzy RM running on the entire blue group directs the support plane to engage in EA against red fighter 1 and the northern red multirole attack plane. In doing so, the support plane protects not only itself, but also the helicopter. The support plane attacks fighter 1 and the multirole attack plane, not the northern LSAR radar. The fuzzy RM directed the support plane to attack the more threatening emitters. This relates to a concept known as "lethality" used by the fuzzy RM to determine a queue of platforms to attack at each time step, and the fact that the support plane has only two EA beams and limited power.

Figure 12 shows that by time step seven the RM running on each member of the blue group has directed them to engage in simultaneous EA against all northern red threats. During time steps eight and nine (not pictured), the RM determines that simultaneous EA against all northern and southern threats is required. Cooperative EA against all threats is initiated in time step nine. For a real-time system, a decision interval is much smaller than the display interval between the time steps indicated at the lower left in Figs. 10 through 13.

Figure 13 indicates that at time step 11 all northern and southern threats are under EA attack by the blue force. Thus, the fuzzy RM has rendered the battlespace secure for the blue group by time step 11.
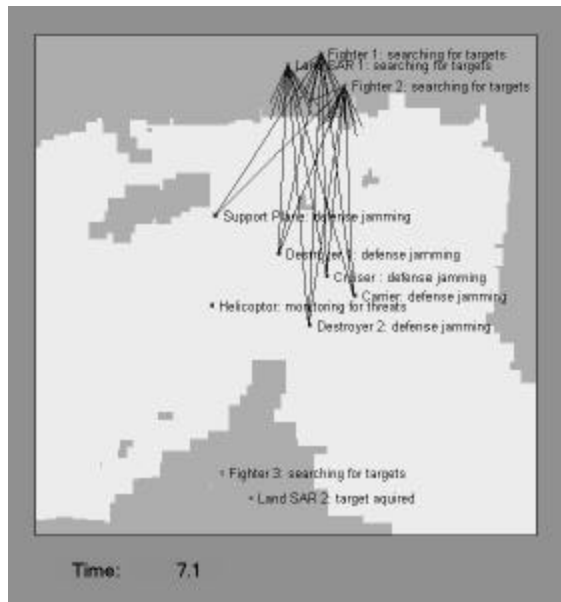

Fig. 12 — By the seventh time step, the RM running on all the blue platforms has directed them to engage in simultaneous EA against all northern red threats.
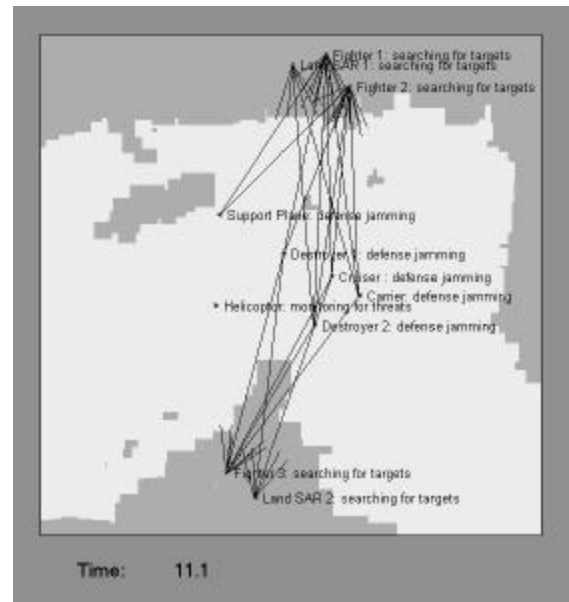

Fig. 13 — By the eleventh time step, the blue RM has secured the battlespace for the blue platform group.

7. MEASURES OF EFFECTIVENESS, COMBINATORIAL EA, AND A GAME-THEORETIC APPROACH TO RULE INVERSION

This section discusses two approaches to the evaluation of algorithms like the RM. The first involves showing large quantities of output, preferably in a graphic or movie format to experts. The second is to construct a measure of effectiveness (MOE) for automatic evaluation of the RM. Also discussed are methods of using this MOE to automatically determine parameters essential to EA techniques through

combinatorial optimization, and an algorithm for automatic inversion of multiplatform doctrine from physic s.

7.1 A Multiplatform MOE

The key features of EA are jamming, disrupting, and deceiving. The broad objective of most EA systems are to deny the enemy the information he seeks, or to surround his return with so much false data that the true information cannot be extracted, or to supply so much false data that the information handling capacity of the victim system is swamped [1].

The MOE described below was developed under the assumption that truth is known as well as the current state and complete history of both red and blue groups. This assumption can be easily satisfied within a digital simulation environment. It is more difficult or impossible to satisfy in a hardware simulation environment due to randomness in physical systems.

The fuzzy MOE for an isolated blue platform engaging a single red platform can be represented as a fuzzy decision tree. It is depicted in Fig. 14. The lowest level boxes or nodes on the tree correspond to root concepts that are described in more detail below. The root concepts are "disrupt," "false target" (FT), "red capable but not firing" (RC), and "lured within blue hard kill range" (LWBHK). The composite concepts are "deny," "delay," and "isolated platform MOE."
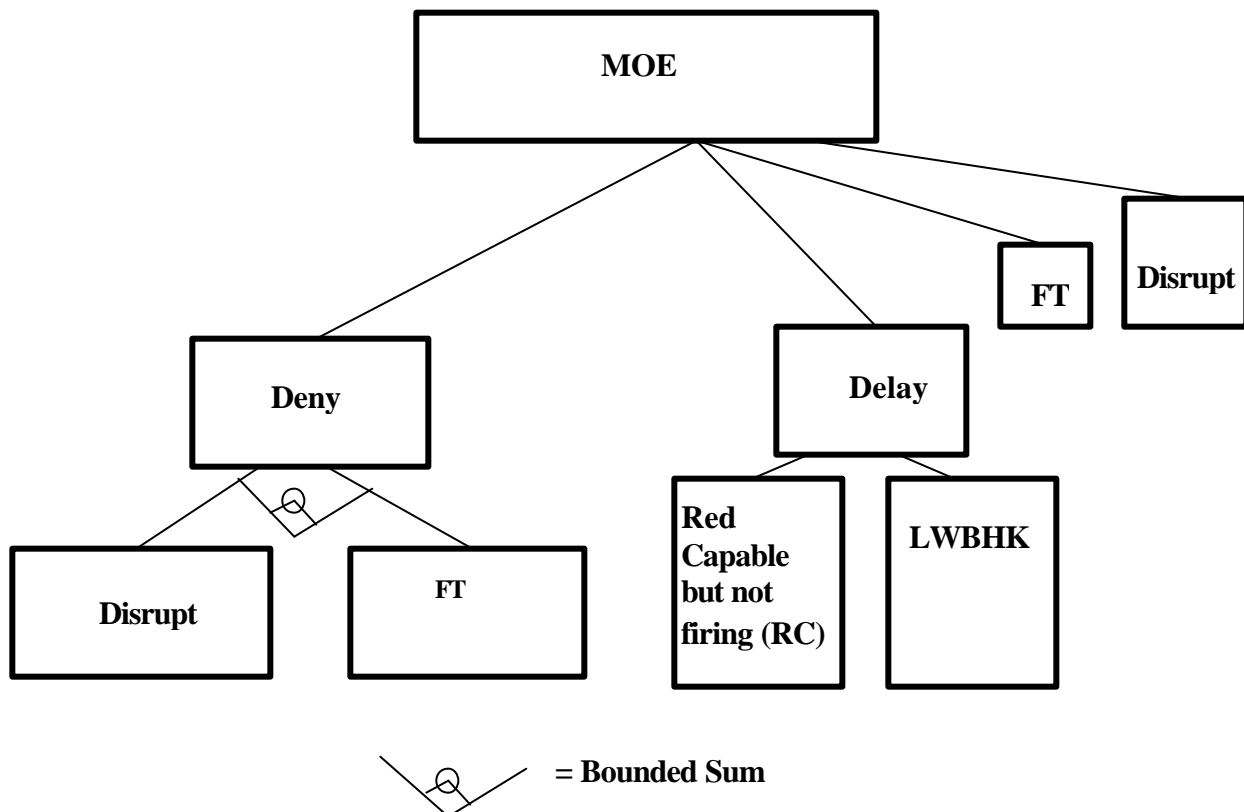

Fig. 14 — The fuzzy MOE tree

The fuzzy grade of membership for the root concept "disrupt" is assigned by first defining a measure of how well a red radar operator can see the blue platform on his radar screen. This measure is the fuzzy grade of membership for the concept "sees-a-target." A measure is defined to determine how "isolated" the pixels of maximum amplitude are on the red radar screen. The fuzzy grade of membership of the root concept "disrupt" is defined to be the degree to which the point of maximum amplitude under the measure "sees-a-target" is "not isolated." The concept "not isolated" is the fuzzy negation of "isolated."

A blue platform is said to have created a false target through EA techniques if it can make a target appear on the red radar operator's screen that does not correspond to a real target. Obviously it is desirable that the false target be some distance away from any real blue target. The fuzzy grade of membership for the root concept "FT" is generally defined as a function of the distance between the real target and the false target.

The root concept "RC" reflects the amount of time red is capable of determining a hardkill solution but does not due to blue's EA efforts. The associated fuzzy grade of membership is usually defined in terms of the difference between the amount of time at each time step that red has not fired a missile and the typical time it takes red to determine a hardkill solution given red's current level of information.

The root concept "LWBHK" relates to how near the red platform is to blue's optimal hardkill range. The associated fuzzy grade of membership is usually defined in terms of the difference between red's range from blue and the optimal range from blue for executing hardkill.

The concept of "deny" represents the ability of the EA technique to deny range information to a red platform. The root concepts "disrupt" and "FT" are combined using a bounded sum [3], i.e., by adding the membership functions for "disrupt" and "FT" and then assigning the sum or unity, whichever is less, to the membership function for the composite concept "deny."

The root concepts "RC" and "LWBHK" are combined through the fuzzy logical connective "or" to give the membership function for the composite concept "delay." Finally concepts "deny," "delay," "FT," and "disrupt" are combined through a fuzzy "or" connective to give the fuzzy grade of membership for the composite concept "MOE."

7.2 Combinatorial EA

An EA technique, i.e., a method of jamming an enemy sensor or creating a false target, is roughly determined by the bandwidth (BW), delay time, duty cycle, and power of the electromagnetic waves used to implement the technique. The above MOE or one similar can be used as a fitness function for a genetic algorithm. Also, by letting the genetic algorithm's chromosomes have traits that correspond to the electromagnetic wave's BW, duty cycle, delay time, and power, it is possible to determine techniques that will result in a maximum value of the MOE. Of course, it is essential to apply engineering constraints. An algorithm of this kind is referred to as a combinatorial EA algorithm. This is a subject of current research.

7.3 Game-theoretic Approach to Automatic Multiplatform Doctrine Inversion from Physics

A global MOE like the one described in Section 7.1 is being investigated as a component of another data mining procedure. Given a group of n-blue platforms and m-red platforms, the EA techniques the blue platforms use at any given time are characterized by their BW, duty cycle, delay time, and power. Assuming these quantities can only take on discrete values, then typically there is a large but finite number of potential combined techniques the n-blue platform group can use. The m-red platform group will typically have a finite number of responses available to them. The different combinations of blue techniques and red responses can be enumerated in a physical simulation environment, allowing crisp if-then statements to be written. Each if-then statement can have an MOE value assigned to it. These MOE values can be interpreted as the elements of a game-theoretic payoff matrix. The game theory problem

can be solved to give probabilities for optimally playing the game. The outcome of such a process yields crisp if-then statements and associated probabilities for each set of blue platform techniques and red platform responses. The resulting database of multiplatform if-then rules and associated probabilities can then be data-mined using a genetic algorithm to create a fuzzy linguistic description. This approach is a current topic of research and is yielding promising results.

8. SUMMARY

A fuzzy logic-based algorithm for optimal allocation and scheduling of electronic attack resources distributed over many platforms is under development. The four decision trees making up the resource manager are discussed. These trees include the isolated and multiplatform decision trees that allow a lone platform or a collection of platforms to respond to a threat. The strategy tree allows the resource manager to make effective use of a threat's past history. The fuzzy parameter selection tree allows threat- and scenario-specific parameterization of fuzzy membership functions. The fuzzy EA model is also discussed. This is an expert system that allows the isolated and multiplatform trees to select the best EA techniques for a given scenario. Examples of the resource manager's multiplatform response are given to illustrate the resource manager's excellent performance.

Optimization of the resource manager (RM) is conducted by using a genetic algorithm as a component of a knowledge discovery process. Construction of the database, which is used for data mining and optimization, is summarized. The approach to optimization is coevolution, a process where both friend and foe agents and meta-agents simultaneously evolve in a complex simulated environment perceived by various sensors. The theory of coevolutionary optimization is introduced, reoptimization criteria and stopping criteria are discussed, an algorithm for automatically constructing coevolutionary fitness functions is introduced, and examples are provided to show the effectiveness of coevolutionary optimization. Examples of the resource manager's multiplatform response are given to illustrate the RM's excellent performance and the value of the information obtained through the knowledge discovery process. Finally, a measure of effectiveness for automatic evaluation of the RM and the information obtained through optimization is discussed.

9. ACKNOWLEDGMENTS

REFERENCES

1. D.C. Schleher, *Electronic Warfare in the Information Age* (Artech House, Boston, 1999), Ch. 1.

2. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems* (Artech House, Boston, 1999), Ch. 11.

3. L.H. Tsoukalas and R.E. Uhrig, *Fuzzy and Neural Approaches in Engineering* (John Wiley and Sons, New York, 1997), Ch. 5.

4. J.H. Holland, *Hidden Order How Adaptation Builds Complexity* (Perseus Books, Reading, 1995), pp. 1-15.

5. H.J. Zimmerman, *Fuzzy Set Theory and its Applications* (Kluwer Academic Publishers Group, Boston, 1991), Ch. 1.

6. J.F. Smith, III and R. Rhyne, II, "A Resource Manager for Distributed Resources: Fuzzy Decision Trees and Genetic Optimization," *Proceedings of the International Conference on Artificial Intelligence, IC-AI'99*, H. Arabnia, ed., Vol. II (CSREA Press, Las Vegas, 1999), pp. 669-675.

7. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, 1989).

8. J.P. Bigus, *Data Mining with Neural Nets* (McGraw-Hill, New York, 1996), Ch. 1.

9. J.C. Bezdek and J.C.Dunn, "Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions," *IEEE Trans. Comp*. **C-24**, 835-838, 1975.

10. J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms* (Plenum Press, New York, 1981).

11. J. Smith, A. Huynh, and M. Kim, "An Application of Clustering and Speed Discrimination to Tracking," NRL/FR/5740--95-9801, Naval Research Laboratory, Washington, DC 20375-5000, December 1995.

12. J.F. Smith III, "A Fuzzy Clustering and Superclustering Scheme for Extracting Structure from Data," NRL/FR/5740--96-9844, Naval Research Laboratory, Washington, DC 20375-5000, December 1996.

13. G.V. Trunk and J.D. Wilson, "Association of DF Bearing Measurements with Radar Track," *IEEE Trans. AES* **AES-23**(4), 438-447, 1987.

14. G.V. Trunk and J.O. Coleman, "Radar-ESM Correction Decision Procedure," NRL Report 8476, Naval Research Laboratory, Washington, DC 20375-5000, May 7, 1981.

15. J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm," *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, ed., Vol. 3068 (SPIE Proceedings, Orlando, 1997), pp. 76-87.

16. J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Theory and Simulation," NRL/FR/5740--97-9866, Naval Research Laboratory, Washington, DC 20375-5000, September 1997.

17. J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Dealing with Multiple Targets, Intermittent Data and Noise," *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, ed., Vol. 3374 (SPIE Proceedings, Orlando, 1998), pp. 2-13.

18. J.F. Smith III, "Fuzzy Logic Association: Performance, Implementation Issues, and Automated Resource Allocation," *Signal Processing, Sensor Fusion, and Target Recognition VIII*, I. Kadar, ed., Vol. 3720 (SPIE Proceedings, Orlando, 1999), pp. 228-238.

19. J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Applied to Noisy, Intermittent and Sparse Data," *Fusion98, Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, R. Hamid and D. Zhu, eds. (CSREA Press, Las Vegas, 1998), pp. 681-688.

20. J.F. Smith III, "A Fuzzy Logic Multisensor Association Algorithm: Multiple Emitters, Computational Complexity, and Noisy Data," NRL Formal Report NRL/FR/5740--99-9918, Naval Research Laboratory, Washington D.C. 20375-5000, July 15, 1999.

21. J.F. Smith III and R. Rhyne II, "Optimal Allocation of Distributed Resources Using Fuzzy Logic and a Genetic Algorithm," NRL Formal Report NRL/FR/5741--00-9970, Naval Research Laboratory, Washington D.C. 20375-5000, September 29, 2000.

22. J.F. Smith III and R.D. Rhyne II, "A Fuzzy Logic Algorithm for Optimal Allocation of Distributed Resources," *Fusion 99: Proceedings of the Second International Conference on Information Fusion*, D. Zhu, ed. (International Society of Information Fusion, San Jose, 1999), pp. 402-409.

23. D. Cliff and G.F. Miller, "Coevolution of Pursuit and Evasion II: Simulation Methods and Results," *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson, eds. (MIT Press Bradford Books, Cambridge, 1996), pp. 1-10.

24. J.F. Smith III and R.D. Rhyne II, "Fuzzy Logic-based Resource Allocation for Isolated and Multiple Platforms," *Signal Processing, Sensor Fusion, and Target Recognition IX*, I. Kadar, ed., Vol. 4052 (SPIE Proceedings, Orlando, 2000), pp. 36-47.

25. J.F. Smith III and R.D. Rhyne II, "Genetic Algorithm-based Optimization of a Fuzzy Logic Resource Manager for Electronic Attack," *Data Mining and Knowledge Discovery II*, B.Dasarathy, ed., Vol. 4057 (SPIE Proceedings, Orlando, 2000), pp. 62-73.

26. J.F. Smith III and R.D. Rhyne II, "A Fuzzy Logic Resource Manager and Underlying Data Mining Techniques," *Fusion2000: Proceedings of the 3rd International Conference on Information Fusion*, R. Reynaud, ed., Vol. II (International Society of Information Fusion, Paris, 2000), pp. WEB1-3 – WEB1-9.

27. J.F. Smith III and R.D. Rhyne II, "Genetic Algorithm Based Optimization of a Fuzzy Logic Resource Manager: Data Mining and Coevolution," *Proceedings of the International Conference on Artificial Intelligence, IC-AI'2000*, H. Arabnia, ed., Vol. I (CSREA Press, Las Vegas, 2000), pp. 421-428.